DeMFI: Deep Joint Deblurring and Multi-Frame Interpolation with Flow-Guided Attentive Correlation and Recursive Boosting

Jihyong Oh[0000−0002−1627−0529] and Munchurl Kim[0000−0003−0146−5419]

Korea Advanced Institute of Science and Technology, Daejeon, South Korea
{jhoh94, mkimee}@kaist.ac.kr

Abstract. We propose a novel joint deblurring and multi-frame interpolation (DeMFI) framework in a two-stage manner, called DeMFI-Net, which converts blurry videos of lower-frame-rate to sharp videos at higher-frame-rate based on flow-guided attentive-correlation-based feature bolstering (FAC-FB) module and recursive boosting (RB), in terms of multi-frame interpolation (MFI). Its baseline version performs feature-flow-based warping with FAC-FB module to obtain a sharp-interpolated frame as well to deblur two center-input frames. Its extended version further improves the joint performance based on pixel-flow-based warping with GRU-based RB. Our FAC-FB module effectively gathers the distributed blurry pixel information over blurry input frames in feature-domain to improve the joint performances. RB trained with recursive boosting loss enables DeMFI-Net to adequately select smaller RB iterations for a faster runtime during inference, even after the training is finished. As a result, our DeMFI-Net achieves state-of-the-art (SOTA) performances for diverse datasets with significant margins compared to recent joint methods. All source codes, including pretrained DeMFI-Net, are publicly available at https://github.com/JihyongOh/DeMFI.

Keywords: Blurry frame interpolation, frame interpolation, deblurring.

1 Introduction

Video frame interpolation (VFI) converts a low frame rate (LFR) video to a high frame rate (HFR) one between given consecutive input frames, thereby providing a visually better motion-smoothed video which is favorably perceived by human visual systems (HVS) [24,25]. Therefore, it is widely used for diverse applications, such as adaptive streaming [50], slow motion generation [18,2,30,28,36,42] and space-time super resolution [22,49,15,48,51,21,52,53,9].

On the other hand, motion blur is necessarily induced by either camera shake [1,56] or object motion [32,57] due to the accumulations of the light during the exposure period [14,16,47] when capturing videos. Therefore, eliminating the motion blur, called deblurring, is essential to synthesize sharp intermediate frames while increasing temporal resolution. The discrete degradation model for

blurriness is generally formulated as follows [20,29,43,19,39,40,13]:

$$\mathbf{B} := \{B_{2i}\}_{i=0,1,\ldots} = \{\frac{1}{2\tau+1}\sum_{j=iK-\tau}^{iK+\tau} S_j\}_{i=0,1,\ldots}, \tag{1}$$

where S_j, $\mathbf{B}$, K and $2\tau+1$ denote latent sharp frame at time j in HFR, observed blurry frames at LFR, a factor that reduces frame rate of HFR to LFR and an exposure time period, respectively. However, a few studies have addressed the joint problem of video frame interpolation with blurred degradation namely as a joint deblurring and frame interpolation problem. To handle this problem effectively, five works [19,39,40,58,13] delicately have shown that joint approach is much better than the cascade of two separate tasks such as deblurring and VFI, which may lead to sub-optimal solutions. However, the methods [19,39,40,13] simply perform a *center*-frame interpolation (CFI) between two blurry center-input frames. This implies that they can only produce intermediate frames of time at a power of 2 in a recursive manner, not for arbitrary time. As a result, prediction errors are accumulatively propagated to the later interpolated frames.

To overcome these limitations for improving the quality in terms of multi-frame interpolation (MFI) with a temporal up-scaling factor $\times M$, we propose a novel framework for joint **De**blurring and **M**ulti-**F**rame **I**nterpolation, called DeMFI-Net, to accurately generate sharp-interpolated frames at arbitrary time t based on flow-guided attentive-correlation-based feature bolstering (FAC-FB) module and recursive boosting (RB). However, using a pretrained optical flow estimator is not optimal for blurry input frames and is computationally heavy. So, our DeMFI-Net is designed to learn *self-induced* feature-flows (f_F) and pixel-flows (f_P) in warping the blurry inputs for synthesizing a sharp-interpolated frame at arbitrary time t, without any pretrained optical flow networks.

Fig. 1: Overview of our DeMFI-Net framework designed in a two-stage manner.

Direct estimation of flows for DeMFI at arbitrary t from the blurry input frames is a very challenging task. To effectively handle it, our DeMFI-Net is designed by a two-stage scheme as shown in Fig. 1: (i) the first stage (baseline version, DeMFI-Net$_{bs}$) jointly performs DeMFI based on *feature-flow-based* warping and blending (FWB) by learning f_F to obtain a sharp-interpolated frame of $t \in (0,1)$ as well to deblur two blurry center-input frames (B_0, B_1) of $t = 0, 1$ from four blurry input frames (B_{-1}, B_0, B_1, B_2), where subscript means a corresponding time index; and (ii) the second stage (recursive boosting,

DeMFI-Net$_{rb}$) further boosts the joint performance based on *pixel-flow-based* warping and blending (PWB) by iteratively updating f_P with the help of GRU-based RB. It is trained with recursive boosting loss that enables to choose smaller iterations for a faster inference during test time, even after the finished training.

On the other hand, the blurry input frames implicitly contain abundant useful latent information due to an accumulation of light [14,16,47], as also shown in Eq. 1. Motivated from this, we propose a novel flow-guided attentive-correlation-based feature bolstering (FAC-FB) module that can effectively bolster the source feature F_0 (or F_1) by extracting the useful information in the feature-domain from its counterpart feature F_1 (or F_0) in guidance of self-induced flow f_{01} (or f_{10}). By doing so, the distributed pixel information over four blurry input frames can be effectively gathered into the corresponding features of the two center-input frames which can then be utilized to pefrom DeMFI effectively.

In the performance evaluation, both two types of DeMFI-Nets outperform previous SOTA methods for three diverse datasets including both various real-world scenes and larger-sized blurry videos with large margins. Extensive experiments with diverse ablation studies have demonstrated the effectiveness of our framework. All source codes including pretrained DeMFI-Net are publicly available at https://github.com/JihyongOh/DeMFI.

2 Related Works

Center-Frame Interpolation (CFI). The VFI methods on CFI only interpolate a *center*-frame between two consecutive *sharp* input frames. CAIN [6] employs a channel attention module to extract motion information effectively. FeFlow [12] adopts deformable convolution [8] in a center frame generator. AdaCoF [26] proposes a warping module in a generalized form to handle motions. However, all the above methods simply do CFI for $\times 2$ increase in frame rates. This approach tends to limit the performance for MFI because they must be recursively applied after each center frame is interpolated, which causes error propagation into later-interpolated frames.

Multi-Frame Interpolation (MFI). To effectively synthesize an intermediate frame at arbitrary time t, many VFI methods on MFI for *sharp* videos adopt a flow-based warping operation. Quadratic video frame interpolation [54,27] adopts the acceleration-aware approximation for the flows in a quadratic form to handle nonlinear motion. DAIN [2] proposes flow projection layer to approximate the flows according to depth information. SoftSplat [31] performs forward warping in feature space with learning-based softmax weights for the occluded region. ABME [35] proposes an asymmetric bilateral motion estimation based on bilateral cost volume [34]. XVFI [42] introduces a recursive multi-scale shared structure to capture extreme motion. However, all the above methods handle MFI problems for *sharp* input frames, which may not work well for *blurry* videos.

Joint Deblurring and Frame Interpolation. The recent studies on the *joint* deblurring and frame interpolation tasks [19,39,40,58,13] have consistently shown that the joint approaches are much better than the simple cascades of two sep-

arately pretrained networks of deblurring and VFI. TNTT [19] first extracts sharp keyframes which are then subsequently used to generate intermediate clear frames by jointly optimizing a cascaded scheme. BIN [39] and its larger-sized version PRF [40] adopts a ConvLSTM-based [41] recurrent pyramid framework to effectively propagate the temporal information over time. ALANET [13] employs the combination of both self- and cross-attention modules to adaptively fuse features in latent spaces. However, all the above four joint methods simply perform the CFI for blurry videos so their performances are limited to MFI. UTI-VFI [58] can interpolate the sharp frames at arbitrary time t in two-stage manner. It first extracts key-state frames, and then warps them to arbitrary time t. However, its performance depends on the quality of flows obtained by a pretrained optical flow network which also increases the complexity (+8.75M parameters).

Distinguished from all the above methods, our proposed framework elaborately learns self-induced f_F and f_P to effectively warp the given blurry input frames for synthesizing a sharp-interpolated frame at arbitrary time, without any pretrained optical flow network. As a result, our method not only outperforms the previous SOTA methods in structural-related metrics but also shows higher *temporal* consistency of visual quality performance for diverse datasets.

3 Proposed Method : DeMFI-Net

Design Considerations. Our proposed DeMFI-Net aims to jointly interpolate a sharp intermediate frame at arbitrary time t and deblur the blurry input frames. Most of the previous SOTA methods [19,40,39,13] only consider CFI ($\times 2$) so need to perform them recursively at the power of 2 for MFI ($\times M$) between two consecutive inputs. Therefore, later-interpolated frames must be *sequentially* created based on their previously-interpolated frames, so the errors are inherently propagated into later-interpolated frames with lower visual qualities. To avoid this, DeMFI-Net is designed to interpolate intermediate frames at multiple time instances without dependency among them. That is, the multiple intermediate frames can be *parallelly* generated. To synthesize an intermediate frame at time $t \in (0, 1)$ instantaneously, we adopt a backward warping [17] which is widely used in VFI research [18,54,2,27,42] to interpolate the frames with estimated flows from time t to 0 and 1, respectively. However, direct usage of a *pretrained* optical flow network is not optimal for blurry frames and even computationally heavy. So our DeMFI-Net is devised to learn self-induced flows in both feature- and pixel-domain via an end-to-end learning. Furthermore, to effectively handle the joint task of deblurring and interpolation, DeMFI-Net is designed in a two-stage manner: baseline version (DeMFI-Net$_{bs}$) and recursive boosting version (DeMFI-Net$_{rb}$) as shown in Fig. 1. DeMFI-Net$_{bs}$ first performs feature-flow-based warping and blending (FWB) in feature-domain where the resulting learned features tend to be more sharply constructed from the blurry inputs. It produces the two deblurred center-inputs and a sharp-interpolated frame at t. Then the output of DeMFI-Net$_{bs}$ is further improved in DeMFI-Net$_{rb}$ via the residual learning, by performing pixel-flow-based warping and blending (PWB).

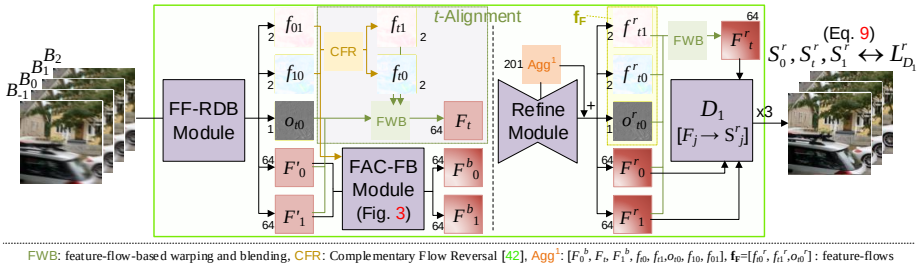


Fig. 2: DeMFI-Net$_{bs}$ based on feature-flows.

3.1 DeMFI-Net$_{bs}$

Fig. 2 shows the architecture of DeMFI-Net$_{bs}$ that first takes four consecutive blurry input frames (B_{-1}, B_0, B_1, B_2). Then, feature flow residual dense backbone (FF-RDB) module is followed which is similar to a backbone network of [40,39], described in *Supplemental*. Its modified 133 ($= 64 \times 2 + 2 \times 2 + 1$) output channels are composed of 64×2 for two feature maps (F'_0, F'_1) followed by tanh functions, 2×2 for two bidirectional feature-domain flows (f_{01}, f_{10}) and 1 for an occlusion map logit (o_{t0}) that is analyzed in detail in *Supplemental*.

t-Alignment. The intermediate flows f_{0t} (or f_{1t}) from time 0 (or 1) to time t are linearly approximated as $f_{0t} = t \cdot f_{01}$ (or $f_{1t} = (1-t) \cdot f_{10}$) based on the f_{01}, f_{10}. Then we apply the complementary flow reversal (CFR) [42] for f_{0t} and f_{1t} to finally approximate f_{t0} and f_{t1}. Finally, we obtain t-aligned feature F_t by applying the backward warping (W_b) [17] for features F'_0, F'_1 followed by a blending operation with the occlusion map. This is called feature-flow-based warping and blending (FWB) (green box in Fig. 2) as follows:

$$\begin{aligned} F_t &= \mathrm{FWB}(F'_0, F'_1, f_{t0}, f_{t1}, o_{t0}, t) \\ &= \frac{(1-t) \cdot \bar{o}_{t0} \cdot W_b(F'_0, f_{t0}) + t \cdot \bar{o}_{t1} \cdot W_b(F'_1, f_{t1})}{(1-t) \cdot \bar{o}_{t0} + t \cdot \bar{o}_{t1}}, \end{aligned} \quad (2)$$

where $\bar{o}_{t0} = \sigma(o_{t0})$ and $\bar{o}_{t1} = 1 - \bar{o}_{t0}$, and σ is a sigmoid activation function.

FAC-FB Module. Since the pixel information is spread over the blurry input frames due to the accumulation of light [14,16,47] as in Eq. 1, we propose a novel FAC-FB module that can effectively bolster the source feature F'_0 (or F'_1) by extracting the useful information in the feature-domain from its counterpart feature F'_1 (or F'_0) in guidance of self-induced flow f_{01} (or f_{10}). The FAC-FB module in Fig. 3 (a) first encodes the two feature maps (F_0, F_1) by passing the outputs (F'_0, F'_1) of the FF-RDB module through its five residual blocks (ResB's). The cascade (**ResB**$^{\times 5}$) of the five ResB's is shared for F'_0 and F'_1.

After obtaining the F_0 and F_1, the flow-guided attentive correlation (FAC) in Fig. 3 (a) computes attentive correlation of F_0 with respect to the positions of its counterpart feature F_1 pointed by the self-induced flow f_{01} as shown in Fig.

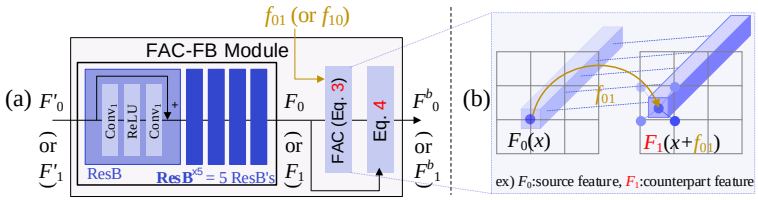


Fig. 3: Flow-guided Attentive Correlation Feature Bolstering (FAC-FB) module.

3 (b). The FAC on F_0 with respect to F_1 guided by f_{01} is calculated as:

$$\begin{aligned}\mathrm{FAC}_{01}(F_0, F_1, f_{01})(\mathbf{x}) = [\ \textstyle\sum_{cw} \mathrm{Conv}_1(F_0(\mathbf{x})) \odot \\ \mathrm{Conv}_1(F_1(\mathbf{x} + f_{01}(\mathbf{x})))]\ \cdot \mathrm{Conv}_1(F_1(\mathbf{x} + f_{01}(\mathbf{x}))),\end{aligned} \quad (3)$$

where $F_1(\mathbf{x} + f_{01}(\mathbf{x}))$ is computed by bilinear sampling on a feature location $\mathbf{x}$. $\odot$, $\sum_{cw}$ and Conv_i denote element-wise multiplication, channel-wise summation and $i \times i$-sized convolution, respectively. The square bracket in Eq. 3 becomes a single-channel scaling map which is then stretched along the channel axis to be element-wise multiplied to $\mathrm{Conv}_1(F_1(\mathbf{x} + f_{01}(\mathbf{x})))$. Finally, the FAC-FB module produces bolstered features F^b_0 for F_0 as:

$$F^b_0 = w_{01} \cdot F_0 + (1 - w_{01}) \cdot \underbrace{\mathrm{Conv}_1(\mathrm{FAC}_{01})}_{\equiv E_0} \quad (4)$$

where w_{01} is a single channel of spatially-variant learnable weights that are dynamically generated by an embedded FAC_{01} via Conv_1 (denoted as E_0) and F_0 according to $w_{01} = (\sigma \circ \mathrm{Conv}_3 \circ \mathrm{ReLU} \circ \mathrm{Conv}_3)([E_0, F_0])$. $[\cdot]$ means a concatenation along a channel axis. Similarly, FAC_{10} and F^b_1 can be computed for F_1 with respect to F_0 by f_{10}. The FAC is computationally efficient because its attentive correlation is only computed in the focused locations pointed by the flows. All filter weights in the FAC-FB module are shared for both F'_0 and F'_1.

Refine Module. After the FAC-FB Module in Fig. 2, F^b_0, F^b_1, f_{t0}, f_{t1} and o_{t0} are refined via the U-Net-based [38] Refine Module (RM) as $[F^r_0, F^r_1, f^r_{t0}, f^r_{t1}, o^r_{t0}] = \mathrm{RM}(\mathbf{Agg}^1) + [F^b_0, F^b_1, f_{t0}, f_{t1}, o_{t0}]$ where $\mathbf{Agg}^1$ is the aggregation of $[F^b_0, F_t, F^b_1, f_{t0}, f_{t1}, o_{t0}, f_{01}, f_{10}]$ in the concatenated form. Then, we get the refined feature F^r_t at time t by $F^r_t = \mathrm{FWB}(F^r_0, F^r_1, f^r_{t0}, f^r_{t1}, o^r_{t0}, t)$ as similar to Eq. 2. Here, we define a composite symbol at time t by the combination of two feature-flows and occlusion map logit as $\mathbf{f_F} \equiv [f^r_{t0}, f^r_{t1}, o^r_{t0}]$ to be used in recursive boosting.

Decoder I (D_1). D_1 is composed of $\mathbf{ResB}^{\times 5}$ and it is designed to have a function: to decode a feature F_j at a time j to a sharp frame S^r_j. D_1 is shared for all the three features (F^r_0, F^r_t, F^r_1). The final sharp outputs of baseline version DeMFI-Net$_{bs}$ are S^r_0, S^r_t and S^r_1 decoded by D_1, which would be applied by L1 reconstruction loss ($L^r_{D_1}$) (Eq. 9). Although DeMFI-Net$_{bs}$ outperforms the previous joint SOTA methods, its extension with recursive boosting, called DeMFI-Net$_{rb}$, can further improve the performance.

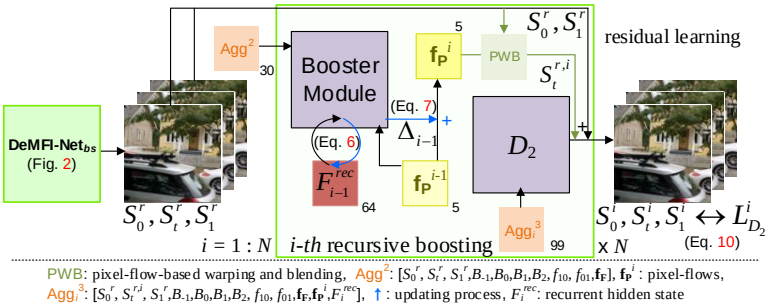


Fig. 4: DeMFI-Net$_{rb}$ at *i-th* Recursive Boosting (RB) based on pixel-flows via residual learning. The operation in the green box can recursively run $N = N_{trn}$ times during training (Eq. 8), and then it can perform $N = N_{tst}$ ($< N_{trn}$) times during testing for faster inference while maintaining high performance.

3.2 DeMFI-Net$_{rb}$

Since we have already obtained sharp output frames S_0^r, S_t^r, S_1^r by DeMFI-Net$_{bs}$, they can further be sharpened based on the pixel-flows by recursive boosting via residual learning. It is known that feature-flows ($\mathbf{f_F}$) and pixel-flows ($\mathbf{f_P}$) would have similar characteristics [26,12]. Therefore, the $\mathbf{f_F}$ obtained from the DeMFI-Net$_{bs}$ are used as initial $\mathbf{f_P}$ for recursive boosting. For this, we design a GRU [5]-based recursive boosting for progressively updating $\mathbf{f_P}$ to perform PWB for two sharp frames at $t = 0, 1$ (S_0^r, S_1^r) accordingly to boost the quality of a sharp intermediate frame at t via residual learning which has been widely adopted for effective deblurring [55,10,37,33,4]. Fig. 4 shows *i-th* recursive boosting (RB) of DeMFI-Net$_{rb}$, which is composed of Booster Module and Decoder II (D_2).

Booster Module. Booster Module iteratively updates $\mathbf{f_P}$ to perform PWB for S_0^r, S_1^r obtained from DeMFI-Net$_{bs}$. The Booster Module is composed of Mixer and GRU-based Booster (GB), and it first takes a recurrent hidden state (F_{i-1}^{rec}) and $\mathbf{f_P}^{i-1}$ at *i-th* recursive boosting as well as an aggregation of several components in the form of $\mathbf{Agg}^2 = [S_0^r, S_t^r, S_1^r, B_{-1}, B_0, B_1, B_2, f_{01}, f_{10}, \mathbf{f_F}]$ as an input to yield two outputs of F_i^{rec} and $\mathbf{\Delta}_{i-1}$ that is added on $\mathbf{f_P}^{i-1}$. Note that $\mathbf{f_P^0} = \mathbf{f_F}$ and $\mathbf{Agg}^2$ is not related to *i-th* recursive boosting. The updating process indicated by blue arrows in Fig. 4 is given as follows:

$$M_{i-1} = \text{Mixer}([\mathbf{Agg}^2, \mathbf{f_P}^{i-1}]), \tag{5}$$

$$[F_i^{rec}, \mathbf{\Delta}_{i-1}] = \text{GB}([F_{i-1}^{rec}, M_{i-1}]), \tag{6}$$

$$\mathbf{f_P}^{i} = \mathbf{f_P}^{i-1} + \mathbf{\Delta}_{i-1}, \tag{7}$$

where the initial feature F_0^{rec} is obtained as a 64-channel feature via channel reduction for $\text{Conv}_1([F_0^r, F_t^r, F_1^r])$ of 192 channels. More details are provided for the Mixer and the updating process of GB in *Supplemental*.

Decoder II (D_2). D_2 in Fig. 4 is composed of $\mathbf{ResB}^{\times 5}$. It fully exploits abundant information of $\mathbf{Agg}_i^3 = [S_0^r, S_t^{r,i}, S_1^r, B_{-1}, B_0, B_1, B_2, f_{01}, f_{10}, \mathbf{f_F}, \mathbf{f_P}^i, F_i^{rec}]$

to generate the refined outputs $[S_0^i, S_t^i, S_1^i] = D_2(\mathbf{Agg}_i^3) + [S_0^r, S_t^{r,i}, S_1^r]$ via residual learning, where $S_t^{r,i} = \text{PWB}(S_0^r, S_1^r, \mathbf{f_P}^i, t)$ is operated by *only* using the updated $\mathbf{f_P}^i$ after the *i-th* RB to enforce the flows to be better boosted.

Loss Functions. The final total loss function $\mathcal{L}_{total}$ for Fig. 1 is given as:

$$\mathcal{L}_{total} = \mathcal{L}_{D_1}^r + \underbrace{\sum_{i=1}^{N_{trn}} \mathcal{L}_{D_2}^i}_{\text{recursive boosting loss}}, \tag{8}$$

$$\mathcal{L}_{D_1}^r = (\textstyle\sum_{j\in(0,t,1)} \|S_j^r - GT_j\|_1)/3, \tag{9}$$

$$\mathcal{L}_{D_2}^i = (\textstyle\sum_{j\in(0,t,1)} \|S_j^i - GT_j\|_1)/3, \tag{10}$$

where GT_j and N_{trn} denote the ground-truth sharp frame at time j and total numbers of RB for training, respectively. We denote DeMFI-Net$_{rb}(N_{trn}, N_{tst})$ as DeMFI-Net$_{rb}$ that is trained with N_{trn} and is tested by N_{tst} recursive boosting. The second term in the right-hand side of Eq. 8 is called as a recursive boosting loss. It should be noted that DeMFI-Net$_{rb}$ is *jointly* trained with the architecture of DeMFI-Net$_{bs}$ in an end-to-end manner using Eq. 8 without any complex learning schedule, and DeMFI-Net$_{bs}$ is trained with only Eq. 9 *from the scratch*.

On the other hand, the design consideration for Booster Module was partially inspired from the work [46] which is here carefully modified for more complex process of DeMFI; (i) Due to the absence of ground-truth for the pixel-flows from t to 0 and 1, *self-induced* pixel-flows are instead learned by adopting D_2 and the recursive boosting loss; (ii) $\mathbf{f_P}$ is not necessarily to be learned precisely, instead to improve the final joint performance of sharpening the S_0^r, S_t^r, S_1^r via PWB and D_2 as shown in Fig. 4. So, we do not block any backpropagation to $\mathbf{f_P}$ per every RB unlike in [46], to fully focus on boosting the performance.

4 Experiments

Training Dataset. To train our network, we use Adobe240 dataset [43] which contains 120 videos of 1,280×720 @ 240fps. We follow a blurry formation setting of [39,40,13] by averaging 11 consecutive frames at a stride of 8 frames over time to synthesize blurry frames captured by a long exposure, which finally generates blurry frames of 30fps with $K = 8$ and $\tau = 5$ in Eq. 1. The resulting blurry frames are downsized to 640×352 as done in [39,40,13].

Implementation Details. Each training sample is composed of four consecutive blurry input frames (B_{-1}, B_0, B_1, B_2) and three sharp-target frames (GT_0, GT_t, GT_1) where t is randomly determined in multiple of 1/8 with $0 < t < 1$ as in [42]. The filter weights of the DeMFI-Net are initialized by the Xavier method [11] and the mini-batch size is set to 2. DeMFI-Net is trained with a total of 420K iterations (7,500 epochs) by using the Adam optimizer [23] with the initial learning rate set to 10^{-4} and reduced by a factor of 2 at the 3,750-, 6,250- and 7,250-*th* epochs. The total numbers of recursive boosting are empirically set to $N_{trn} = 5$ for training and $N_{tst} = 3$ for testing. We construct each training sample on the fly by randomly cropping a 256 × 256-sized patch from blurry and clean frames, and it is randomly flipped in both spatial and temporal directions

for data augmentation. Training takes about five days for DeMFI-Net$_{bs}$ and ten days for DeMFI-Net$_{rb}$ by using an NVIDIA RTX™ GPU with PyTorch.

4.1 Comparison to Previous SOTA Methods

We mainly compare our DeMFI-Net with five previous joint SOTA methods; TNTT [19], UTI-VFI [58], BIN [39], PRF [40] (a larger-sized version of BIN) and ALANET [13], which all have adopted joint learning for deblurring and VFI. They all have reported better performance than the cascades of separately trained VFI [18,3,2] and deblurring [45,49] networks. It should be noted that the four methods of TNTT, BIN, PRF and ALANET simply perform CFI (×2), not at arbitrary t but at the center time $t = 0.5$. So, they have to perform MFI (×8) recursively based on previously interpolated frames, which causes to propagate interpolation errors into later-interpolated frames. For experiments, we delicately compare them in two aspects of CFI and MFI. For MFI performance, *temporal* consistency is measured such that the pixel-wise difference of motions are calculated in terms of tOF [7,42] (the lower, the better) for all 7 interpolated frames and deblurred two center frames for each blurry test sequence (scene). We also retrain the UTI-VFI with the same blurry formation setting [39,40,13] for the Adobe240 for fair comparison, to be denoted as UTI-VFI*.

Test Dataset. We use three datasets for evaluation: (i) Adobe240 dataset [43], (ii) YouTube240 dataset and (iii) GoPro240 dataset (CC BY 4.0 license) [29] that contains large dynamic object motions and camera shakes. For the YouTube240, we directly selected 60 YouTube videos of 1,280×720 at 240fps by considering to include extreme scenes captured by diverse devices. Then they were resized to 640×352 as done in [39,40,13]. The Adobe240 contains 8 videos of 1,280×720 resolution at 240 fps and was also resized to 640×352, which is totally composed of 1,303 blurry input frames. On the other hand, the GoPro240 has 11 videos with total 1,500 blurry input frames but we used the original size of 1,280×720 for an extended evaluation in larger-sized resolution. Please note that all test datasets are also temporally downsampled to 30 fps with the blurring as [39,40,13].

Quantitative Comparison. Table 1 shows the quantitative performance comparisons for the previous SOTA methods including the cascades of deblurring and VFI methods with the Adobe240, in terms of deblurring and CFI (×2). Most results of the previous methods in Table 1 are brought from [39,40,13], except those of UTI-VFI (*pretrained, newly tested*), UTI-VFI* (*retrained, newly tested*) and DeMFI-Nets (ours). Please note that all runtimes (R$_t$) in Table 1 were measured for 640×352-sized frames in the setting of [39,40] with one NVIDIA RTX™ GPU. As shown in Table 1, our proposed DeMFI-Net$_{bs}$ and DeMFI-Net$_{rb}$ clearly outperform all the previous methods with large margins in both deblurring and CFI performances, and the number of model parameters (#P) for our methods are the second- and third-smallest with smaller R$_t$ compared to PRF. In particular, DeMFI-Net$_{rb}$(5,3) outperforms ALANET by 1dB and 0.0093 in terms of PSNR and SSIM, respectively for average performances of deblurring and CFI, and especially by average 1.51dB and 0.0124 for center-interpolated frames attributed to our warping-based framework with self-induced flows. Furthermore,

Table 1: Quantitative comparisons on Adobe240 [43] for deblurring and center-frame interpolation (×2).

Method	R_t (s)	#P (M)	Deblurring		CFI (×2)		Average	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
B_0, B_1	-	-	28.68	0.8584	-	-	-	-
SloMo [18]	-	39.6	-	-	27.52	0.8593	-	-
MEMC [3]	-	70.3	-	-	30.83	0.9128	-	-
DAIN [2]	-	24.0	-	-	31.03	0.9172	-	-
SRN [45]+[18]	0.27	47.7			27.22	0.8454	28.32	0.8604
SRN [45]+[3]	0.22	78.4	29.42	0.8753	28.25	0.8625	28.84	0.8689
SRN [45]+[2]	0.79	32.1			27.83	0.8562	28.63	0.8658
EDVR [49]+[18]	0.42	63.2			27.79	0.8671	30.28	0.9003
EDVR [49]+[3]	0.27	93.9	32.76	0.9335	30.22	0.9058	31.49	0.9197
EDVR [49]+[2]	1.13	47.6			30.28	0.9070	31.52	0.9203
UTI-VFI [58]	0.80	43.3	28.73	0.8656	29.00	0.8690	28.87	0.8673
UTI-VFI*	0.80	43.3	31.02	0.9168	32.67	0.9347	31.84	0.9258
TNTT [19]	0.25	10.8	29.40	0.8734	29.24	0.8754	29.32	0.8744
BIN [39]	0.28	4.68	32.67	0.9236	32.51	0.9280	32.59	0.9258
PRF [40]	0.76	11.4	33.33	0.9319	33.31	0.9372	33.32	0.9346
ALANET [13]	-	-	33.71	0.9329	32.98	0.9362	33.34	0.9355
DeMFI-Net$_{bs}$	0.38	5.96	33.83	0.9377	33.93	0.9441	33.88	0.9409
DeMFI-Net$_{rb}$(1,1)	0.51	7.41	34.06	0.9401	34.35	0.9471	34.21	0.9436
DeMFI-Net$_{rb}$(5,3)	0.61	7.41	**34.19**	**0.9410**	**34.49**	**0.9486**	**34.34**	**0.9448**

RED: Best performance, BLUE: Second best performance.
R_t: The runtime on 640×352-sized frames (s), UTI-VFI*: retrained version.
#P: The number of parameters (M), ALANET: no source code for testing.

even our DeMFI-Net$_{bs}$ is superior to all previous methods which are dedicatedly trained for CFI.

Table 2 shows quantitative comparisons of the joint methods for the three test datasets in terms of deblurring and MFI (×8). As shown in Table 2, all the three versions of DeMFI-Net significantly outperform the previous joint methods, which shows a good generalization of our DeMFI-Net framework. Fig. 5 shows PSNR profiles for MFI results (×8). As shown, the CFI methods such as TNTT and PRF tend to synthesize worse intermediate frames than the methods of interpolation at arbitrary time like UTI-VFI and our DeMFI-Net. This is because the error propagation is accumulated recursively due to the inaccurate interpolations by the CFI methods, which also has been inspected in VFI for sharp input frames [42]. On the other hand, we also recursively do CFI (×2) three times to measure *sequential* inference performances of DeMFI-Net$_{rb}$(5,3), indicated by 'DeMFI-seq_' of pink color in Fig. 5, which also clearly shows that the errors are accumulatively propagated into the later interpolated frames.

Although UTI-VFI can interpolate the frames at arbitrary t by adopting the PWB combined with QVI [54], its performances inevitably depend on f_P quality obtained by PWC-Net [44], where adoption of a pretrained net brings a disadvantage in terms of both R_t and #P (+8.75M). It is worthwhile to note that our method also shows the best performances in terms of temporal consistency with tOF by help of *self-induced* flows in interpolating frames at arbitrary t.

Qualitative Comparison. Fig. 6 shows the visual comparisons of deblurring and VFI performances on YouTube240 and GoPro240 datasets, respectively. As

Table 2: Quantitative comparisons of joint methods on Adobe240 [43], YouTube240 and GoPro240 [29] datasets for deblurring and multi-frame interpolation (×8). R_t and FLOPS are measured on 640×352-sized frames.

Joint Method	Adobe240 [43]			YouTube240		
	deblurring PSNR/SSIM	MFI (×8) PSNR/SSIM	Average PSNR/SSIM/tOF	deblurring PSNR/SSIM	MFI (×8) PSNR/SSIM	Average PSNR/SSIM/tOF
UTI-VFI [58]	28.73/0.8657	28.66/0.8648	28.67/0.8649/0.578	28.61/0.8891	28.64/0.8900	28.64/0.8899/0.585
UTI-VFI*	31.02/0.9168	32.30/0.9292	32.13/0.9278/**0.445**	30.40/0.9055	31.76/0.9183	31.59/0.9167/0.517
TNTT [19]	29.40/0.8734	29.45/0.8765	29.45/0.8761/0.559	29.59/0.8891	29.77/0.8901	29.75/0.8899/0.549
PRF [40]	33.33/0.9319	28.99/0.8774	29.53/0.8842/0.882	32.37/0.9199	29.11/0.8919	29.52/0.8954/0.771
DeMFI-Net$_{bs}$	33.83/0.9377	33.79/0.9410	33.79/0.9406/0.473	32.90/0.9251	32.79/0.9262	32.80/0.9260/0.469
DeMFI-Net$_{rb}$(1,1)	34.06/0.9401	34.15/0.9440	34.14/0.9435/0.460	33.17/0.9266	33.22/0.9291	33.21/0.9288/**0.459**
DeMFI-Net$_{rb}$(5,3)	**34.19**/**0.9410**	**34.29**/**0.9454**	**34.28**/**0.9449**/0.457	**33.31**/**0.9282**	**33.33**/**0.9300**	**33.33**/**0.9298**/0.461

Joint Method	R_t(s)	#P(M)	FLOPS	GoPro240 [29]		
				deblurring PSNR/SSIM	MFI (×8) PSNR/SSIM	Average PSNR/SSIM/tOF
UTI-VFI [58]	0.80	43.3	3.23T	25.66/0.8085	25.63/0.8148	25.64/0.8140/0.716
UTI-VFI*	0.80	43.3	3.23T	28.51/0.8656	29.73/0.8873	29.58/0.8846/0.558
TNTT [19]	0.25	10.8	609.62G	26.48/0.8085	26.68/0.8148	26.65/0.8140/0.754
PRF [40]	0.76	11.4	3.2T	30.27/0.8866	25.68/0.8053	26.25/0.8154/1.453
DeMFI-Net$_{bs}$	0.38	5.96	748.57G	30.54/0.8935	30.78/0.9019	30.75/0.9008/0.538
DeMFI-Net$_{rb}$(1,1)	0.51	7.41	1.07T	30.63/0.8961	31.10/0.9073	31.04/0.9059/0.512
DeMFI-Net$_{rb}$(5,3)	0.61	7.41	1.71T	**30.82**/**0.8991**	**31.25**/**0.9102**	**31.20**/**0.9088**/**0.500**

shown, the blurriness is easily visible between B_0 and B_1, which is challenging for VFI. Our DeMFI-Nets show better generalized performances for the extreme scenes (Fig. 6 (a)) and larger-sized videos (Fig. 6 (b)), also in terms of temporal consistency. Due to page limits, more visual comparisons with larger sizes are provided in *Supplemental* for all three test datasets. Also the results of deblurring and MFI (×8) of all the SOTA methods are publicly available at https://github.com/JihyongOh/DeMFI. Please note that it is laborious but worth to get results for the SOTA methods in terms of MFI (×8).

Table 3: Ablation experiments on RB and FAC in terms of total *average* of deblurring and MFI (×8); 'w/o FAC' means $F_0^b = F_0$.

Method	R_t (s)	#P (M)	Adobe240 PSNR	Adobe240 SSIM	YouTube240 PSNR	YouTube240 SSIM
(a) w/o RB, w/o FAC	0.32	5.87	33.30	0.9361	32.54	0.9230
(b) w/o RB, $f = 0$	0.38	5.96	33.64	0.9393	32.74	0.9237
(c) w/o RB (DeMFI-Net$_{bs}$)	0.38	5.96	33.79	0.9406	32.80	0.9260
(d) w/o FAC	0.45	7.32	33.73	0.9391	32.93	0.9260
(e) $f = 0$	0.51	7.41	34.08	0.9428	33.15	0.9279
(f) DeMFI-Net$_{rb}$(1,1)	0.51	7.41	**34.14**	**0.9435**	**33.21**	**0.9288**

Table 4: Ablation study on N_{trn} and N_{tst} of DeMFI-Net$_{rb}$.

N_{trn} \ N_{tst}	PSNR/SSIM		
	1 ($R_t = 0.51$)	3 ($R_t = 0.61$)	5 ($R_t = 0.68$)
1	**34.14**/**0.9435**	28.47/0.8695	25.99/0.8136
	33.21/**0.9288**	29.01/0.8845	26.56/0.8406
3	**34.21**/0.9439	**34.21**/**0.9440**	34.16/0.9437
	33.27/0.9290	**33.27**/**0.9291**	33.23/0.9289
5	34.27/0.9446	**34.28**/**0.9449**	34.27/0.9448
	33.32/0.9296	**33.33**/**0.9298**	**33.33**/0.9297

1st/2nd row: Adobe240/YouTube240 in each block.
RED: Best performance of each row, #P=7.41M.

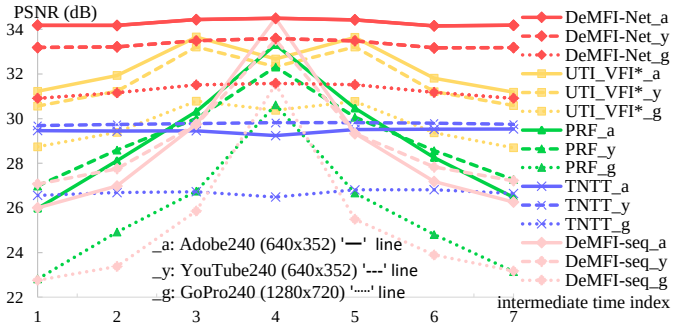
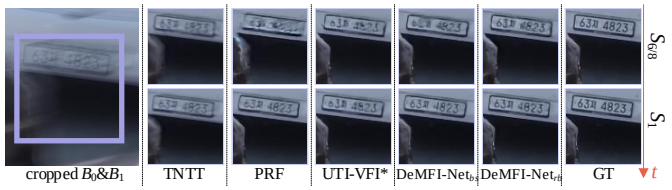


Fig. 5: PSNR profiles for multi-frame interpolation results (×8) for the *blurry* input frames on diverse three datasets; Adobe240, YouTube240 and GoPro240. The number of horizontal axis is the intermediate time index between two blurry center-input frames (0, 8). Our DeMFI-Net$_{rb}$(5,3), indicated by 'DeMFI-Net_' of red color, consistently shows best performances along all time instances.

(a) YouTube240



(b) GoPro240

Fig. 6: Visual comparisons for MFI results of our DeMFI-Nets and joint SOTA methods on (a) YouTube240 and (b) GoPro240. *Best viewed in zoom.* Demo video is available at https://youtu.be/J93tW1uwRy0.

4.2 Ablation Studies

To analyze the effectiveness of each component in our framework, we perform ablation experiments. Table 3 shows the results of ablation experiments for FAC in Fig. 3 and RB in Fig. 4 with $N_{trn} = 1$ and $N_{tst} = 1$ for a simplicity.

FAC. By comparing the method (f) to (d) and (c) to (a) in Table 3, it is noticed that FAC can effectively improve the overall joint performances in the both cases without and with RB by taking little more runtime (+0.06s) and small number of additional parameters (+0.09M). Fig. 7 qualitatively shows the effect of FAC for DeMFI-Net$_{rb}$(1,1) (f). Brighter positions with green boxes in the rightmost column indicate important regions E_1 after passing Eq. 3 and Conv$_1$. The green boxes show blurrier patches that are more attentive in the counterpart feature based on f_{10} to reinforce the source feature F_1 complementally. On the other hand, the less focused regions such as backgrounds with less blurs are relatively have smaller E after FAC. In summary, FAC bolsters the source feature by complementing the important regions with blurs in the counterpart feature pointed by flow-guidance. We also show the effectiveness of FAC without flow guidance when trained with $f = 0$. As shown in Table 3, we obtained the performance higher than without FAC but lower than with FAC by flow-guidance, as expected. Therefore, we conclude that FAC works very effectively under the self-induced flow guidance to bolster the center features to improve the performance of the joint task.

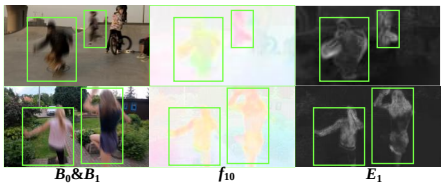


Fig. 7: Effect of FAC. The green boxes show blurrier patches that are more attentive in the counterpart feature based on flow-guidance to effectively bolster the source feature.

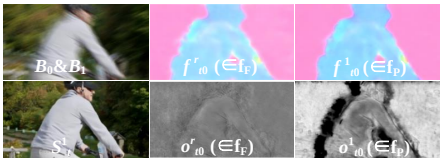


Fig. 8: Self-induced flows for both features $\mathbf{f_F}$ and images $\mathbf{f_P}$ ($t = 7/8$) of DeMFI-Net$_{rb}$ (1,1) show a similar tendency. *They do not have to be accurate, but help improve final joint performances.*

Recursive Boosting. By comparing the method (d) to (a), (e) to (b) and (f) to (c) in Table 3, it can be known that the RB consistently yields improved final joint results. Fig. 8 shows that $\mathbf{f_F}$ and $\mathbf{f_P}$ have a similar tendency in flow characteristics. Furthermore, the $\mathbf{f_P}$ updated from $\mathbf{f_F}$ seems sharper to perform PWB in pixel domain, which may help our two-stage approach effectively handles the joint task based on warping operation. It is noted that our weakest variant (a) (w/o both RB and FAC) even outperformed the second-best joint method (UTI-VFI*) as shown in Table 2, 3 on the both Adobe240 and YouTube240.

# of Recursive Boosting N. To inspect the relationship between N_{trn} and N_{tst} for RB, we train three variants of DeMFI-Net$_{rb}$ each for $N_{trn} = 1, 3, 5$ as shown in Table 4. Since the weight parameters in RB are shared for each recursive boosting, all the variants have same #P=7.41M and each column in Table 4 has same runtime R_t. The performances are generally boosted by increasing N_{trn}, where each recursion is attributed to the recursive boosting loss that enforces the recursively updated flows $\mathbf{f_P}^i$ to better focus on synthesis $S_t^{r,i}$ via the PWB. It should be noted that the overall performances are better when $N_{tst} \leq N_{trn}$, while they are dropped otherwise. So, we can adequately choose smaller N_{tst} ($\leq N_{trn}$) for a faster runtime by considering computational constraints while maintaining high performances, even though the training with N_{trn} is once over. That is, under the same runtime constraint of each R_t as in the column of Table 4 when testing, we can also select the model trained with larger N_{trn} to obtain better results. On the other hand, we found out that further increasing N_{trn} does not bring additional benefits due to saturated performance of DeMFI-Net$_{rb}$.

Extensibility of FAC-FB module and RB. Both FAC-FC module in Fig. 3 and RB in Fig. 4 can be easily inserted in a flow-based network to boost its performance for a specific task. To show the extensibility for our two proposed modules, we trained two variants of the SOTA VFI method for sharp videos, XVFI-Net [42], using default training conditions in their official code by inserting (i) FAC-FB module in front of BIOF-T [42], and (ii) RB behind BIOF-T. We obtained 0.08 dB PSNR gain for the FAC-FB module and 0.07 dB gain for RB ($N_{trn} = 2$, $N_{tst} = 2$) on X-TEST test dataset [42] with $S_{tst} = 3$ [42]. This shows that FAC-FB module and RB can be inserted in *flow-based* network architectures to boost performance, showing extensibility and generalization ability of the proposed modules.

5 Conclusion

We propose a novel joint deblurring and multi-frame interpolation framework in a two-stage manner, called DeMFI-Net, based on our novel flow-guided attentive-correlation-based feature bolstering (FAC-FB) module and recursive boosting (RB), by learning the self-induced feature- and pixel-domain flows without any help of pretrained optical flow networks. FAC-FB module forcefully enriches the source feature by extracting attentive correlation from the counterpart feature at the position where self-induced feature-flow points at, to finally improve results for the joint task. RB trained with recursive boosting loss enables DeMFI-Net to adequately select smaller RB iterations for a faster runtime during inference while keeping performances, even after the training is finished. Our DeMFI-Net achieves state-of-the-art joint performances for diverse datasets with significant margins compared to the previous joint SOTA methods.

Acknowledgement. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00419, Intelligent High Realistic Visual Processing for Smart Broadcasting Media).

References

1. Bahat, Y., Efrat, N., Irani, M.: Non-uniform blind deblurring by reblurring. In: ICCV. pp. 3286–3294 (2017) 1
2. Bao, W., Lai, W.S., Ma, C., Zhang, X., Gao, Z., Yang, M.H.: Depth-aware video frame interpolation. In: CVPR. pp. 3703–3712 (2019) 1, 3, 4, 9, 10
3. Bao, W., Lai, W.S., Zhang, X., Gao, Z., Yang, M.H.: Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. IEEE transactions on pattern analysis and machine intelligence (2019) 9, 10
4. Chi, Z., Wang, Y., Yu, Y., Tang, J.: Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning. In: CVPR. pp. 9137–9146 (2021) 7
5. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. In: EMNLP 7
6. Choi, M., Kim, H., Han, B., Xu, N., Lee, K.M.: Channel attention is all you need for video frame interpolation. In: AAAI. pp. 10663–10671 (2020) 3
7. Chu, M., You, X., Jonas, M., Laura, L.T., Nils, T.: Learning temporal coherence via self-supervision for gan-based video generation. ACM ToG **39**(4), 75–1 (2020) 9
8. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: CVPR. pp. 764–773 (2017) 3
9. Dutta, S., Shah, N.A., Mittal, A.: Efficient space-time video super resolution using low-resolution flow and mask upsampling. In: CVPR. pp. 314–323 (2021) 1
10. Gao, H., Tao, X., Shen, X., Jia, J.: Dynamic scene deblurring with parameter selective sharing and nested skip connections. In: CVPR. pp. 3848–3856 (2019) 7
11. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS. pp. 249–256 (2010) 8
12. Gui, S., Wang, C., Chen, Q., Tao, D.: Featureflow: Robust video interpolation via structure-to-texture generation. In: CVPR. pp. 14004–14013 (2020) 3, 7
13. Gupta, A., Aich, A., Roy-Chowdhury, A.K.: Alanet: Adaptive latent attention network for joint video deblurring and interpolation. In: ACMMM. pp. 256–264 (2020) 2, 3, 4, 8, 9, 10
14. Gupta, A., Joshi, N., Zitnick, C.L., Cohen, M., Curless, B.: Single image deblurring using motion density functions. In: ECCV. pp. 171–184. Springer (2010) 1, 3, 5
15. Haris, M., Shakhnarovich, G., Ukita, N.: Space-time-aware multi-resolution video enhancement. In: CVPR. pp. 2859–2868 (2020) 1
16. Harmeling, S., Michael, H., Schölkopf, B.: Space-variant single-image blind deconvolution for removing camera shake. NeurIPS **23**, 829–837 (2010) 1, 3, 5
17. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: NeurIPS. pp. 2017–2025 (2015) 4, 5
18. Jiang, H., Sun, D., Jampani, V., Yang, M.H., Learned-Miller, E., Kautz, J.: Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In: CVPR. pp. 9000–9008 (2018) 1, 4, 9, 10
19. Jin, M., Hu, Z., Favaro, P.: Learning to extract flawless slow motion from blurry videos. In: CVPR. pp. 8112–8121 (2019) 2, 3, 4, 9, 10, 11
20. Jin, M., Meishvili, G., Favaro, P.: Learning to extract a video sequence from a single motion-blurred image. In: CVPR (June 2018) 2
21. Kang, J., Jo, Y., Oh, S.W., Vajda, P., Kim, S.J.: Deep space-time video upsampling networks. In: ECCV. pp. 701–717. Springer (2020) 1

22. Kim, S.Y., Oh, J., Kim, M.: Fisr: Deep joint frame interpolation and super-resolution with a multi-scale temporal loss. In: AAAI. pp. 11278–11286 (2020) 1
23. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015) 8
24. Kuroki, Y., Nishi, T., Kobayashi, S., Oyaizu, H., Yoshimura, S.: A psychophysical study of improvements in motion-image quality by using high frame rates. Journal of the Society for Information Display **15**(1), 61–68 (2007) 1
25. Kuroki, Y., Takahashi, H., Kusakabe, M., Yamakoshi, K.i.: Effects of motion image stimuli with normal and high frame rates on eeg power spectra: comparison with continuous motion image stimuli. Journal of the Society for Information Display **22**(4), 191–198 (2014) 1
26. Lee, H., Kim, T., Chung, T.y., Pak, D., Ban, Y., Lee, S.: Adacof: Adaptive collaboration of flows for video frame interpolation. In: CVPR. pp. 5316–5325 (2020) 3, 7
27. Liu, Y., Xie, L., Siyao, L., Sun, W., Qiao, Y., Dong, C.: Enhanced quadratic video interpolation. In: ECCV. pp. 41–56. Springer (2020) 3, 4
28. Liu, Z., Yeh, R.A., Tang, X., Liu, Y., Agarwala, A.: Video frame synthesis using deep voxel flow. In: CVPR. pp. 4463–4471 (2017) 1
29. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: CVPR. pp. 3883–3891 (2017) 2, 9, 11
30. Niklaus, S., Liu, F.: Context-aware synthesis for video frame interpolation. In: CVPR. pp. 1701–1710 (2018) 1
31. Niklaus, S., Liu, F.: Softmax splatting for video frame interpolation. In: CVPR. pp. 5437–5446 (2020) 3
32. Pan, J., Sun, D., Pfister, H., Yang, M.H.: Blind image deblurring using dark channel prior. In: CVPR. pp. 1628–1636 (2016) 1
33. Park, D., Kang, D.U., Kim, J., Chun, S.Y.: Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In: ECCV. pp. 327–343. Springer (2020) 7
34. Park, J., Ko, K., Lee, C., Kim, C.S.: Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In: ECCV (2020) 3
35. Park, J., Lee, C., Kim, C.S.: Asymmetric bilateral motion estimation for video frame interpolation. In: ICCV (2021) 3
36. Peleg, T., Szekely, P., Sabo, D., Sendik, O.: Im-net for high resolution video frame interpolation. In: CVPR. pp. 2398–2407 (2019) 1
37. Purohit, K., Rajagopalan, A.: Region-adaptive dense network for efficient motion deblurring. In: AAAI. vol. 34, pp. 11882–11889 (2020) 7
38. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015) 6
39. Shen, W., Bao, W., Zhai, G., Chen, L., Min, X., Gao, Z.: Blurry video frame interpolation. In: CVPR. pp. 5114–5123 (2020) 2, 3, 4, 5, 8, 9, 10
40. Shen, W., Bao, W., Zhai, G., Chen, L., Min, X., Gao, Z.: Video frame interpolation and enhancement via pyramid recurrent framework. IEEE Transactions on Image Processing **30**, 277–292 (2020) 2, 3, 4, 5, 8, 9, 10, 11
41. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: NeurIPS (2015) 4
42. Sim, H., Oh, J., Kim, M.: Xvfi: extreme video frame interpolation. In: ICCV (2021) 1, 3, 4, 5, 8, 9, 10, 14

43. Su, S., Delbracio, M., Wang, J., Sapiro, G., Heidrich, W., Wang, O.: Deep video deblurring for hand-held cameras. In: CVPR. pp. 1279–1288 (2017) 2, 8, 9, 10, 11
44. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: CVPR. pp. 8934–8943 (2018) 10
45. Tao, X., Gao, H., Shen, X., Wang, J., Jia, J.: Scale-recurrent network for deep image deblurring. In: CVPR. pp. 8174–8182 (2018) 9, 10
46. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: ECCV. pp. 402–419. Springer (2020) 8
47. Telleen, J., Sullivan, A., Yee, J., Wang, O., Gunawardane, P., Collins, I., Davis, J.: Synthetic shutter speed imaging. In: Computer Graphics Forum. vol. 26, pp. 591–598. Wiley Online Library (2007) 1, 3, 5
48. Tian, Y., Zhang, Y., Fu, Y., Xu, C.: Tdan: Temporally-deformable alignment network for video super-resolution. In: CVPR. pp. 3360–3369 (2020) 1
49. Wang, X., Chan, K.C., Yu, K., Dong, C., Change Loy, C.: Edvr: Video restoration with enhanced deformable convolutional networks. In: CVPRW. pp. 0–0 (2019) 1, 9, 10
50. Wu, J., Yuen, C., Cheung, N.M., Chen, J., Chen, C.W.: Modeling and optimization of high frame rate video transmission over wireless networks. IEEE Transactions on Wireless Communications **15**(4), 2713–2726 (2015) 1
51. Xiang, X., Tian, Y., Zhang, Y., Fu, Y., Allebach, J.P., Xu, C.: Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution. In: CVPR. pp. 3370–3379 (2020) 1
52. Xiao, Z., Xiong, Z., Fu, X., Liu, D., Zha, Z.J.: Space-time video super-resolution using temporal profiles. In: ACM MM. pp. 664–672 (2020) 1
53. Xu, G., Xu, J., Li, Z., Wang, L., Sun, X., Cheng, M.M.: Temporal modulation network for controllable space-time video super-resolution. In: CVPR. pp. 6388–6397 (2021) 1
54. Xu, X., Siyao, L., Sun, W., Yin, Q., Yang, M.H.: Quadratic video interpolation. In: NeurIPS. pp. 1647–1656 (2019) 3, 4, 10
55. Zhang, H., Dai, Y., Li, H., Koniusz, P.: Deep stacked hierarchical multi-patch network for image deblurring. In: CVPR. pp. 5978–5986 (2019) 7
56. Zhang, K., Luo, W., Zhong, Y., Ma, L., Liu, W., Li, H.: Adversarial spatio-temporal learning for video deblurring. IEEE Transactions on Image Processing **28**(1), 291–301 (2018) 1
57. Zhang, K., Luo, W., Zhong, Y., Ma, L., Stenger, B., Liu, W., Li, H.: Deblurring by realistic blurring. In: CVPR. pp. 2737–2746 (2020) 1
58. Zhang, Y., Wang, C., Tao, D.: Video frame interpolation without temporal priors. NeurIPS **33** (2020) 2, 3, 4, 9, 10, 11